



TECHNISCHE UNIVERSITÄT CHEMNITZ

Fakultät Informatik

Medieninformatik

Studienarbeit

Evaluierung verschiedener Ansätze zur Indexierung im
inhaltsbasierten Image Retrieval

Stefanie Daßler

Chemnitz, den 29. Januar 2008

Prüfer: Professor Dr. Maximilian Eibl

Daßler, Stefanie

Evaluierung verschiedener Ansätze zur Indexierung im inhaltsbasierten Image Retrieval

Studienarbeit, Fakultät Informatik

Technische Universität Chemnitz, Januar 2008

Inhaltsverzeichnis

1	Image Retrieval	1
1.1	Bedeutung des Image Retrieval	1
1.2	Ausblick	1
2	Grundlagen	3
2.1	RGB-Farbmodell	3
2.2	Indizierte Farbtabelle	4
2.3	Vorteile der indizierten Farbtabelle	5
2.4	Erstellung eines Histogramms	5
2.4.1	Angulare und hybride Histogramme	6
2.4.2	Annulare Histogramme	7
2.4.3	Nutzung des annularen Histogramms	8
3	Distanzklassifikation	9
3.1	Bildklassifikation	9
3.1.1	Euler Distanz	9
3.1.2	Histogramm Distanz	9
3.1.3	Distanzmetrik	10
4	Indexierung mit Hilfe des Evolving Tree	12
4.1	Probleme bei üblichen neuronalen Netzen und Self-Organizing Maps	12
4.2	Definition des Evolving Trees	13
4.3	Evolving Tree Algorithmus	13
4.4	Evolving Tree als Index	15
5	Vantage Point Tree als Indexstruktur	18
5.1	Definition des Vantage Point Tree	18
5.2	Algorithmus des Vantage Point Tree	18
5.2.1	N-Nearest-Neighbor im VP-Baum	18
5.2.2	Auswahl des Vantage Point	20
5.2.3	Persistenzproblem	20
5.3	Umsetzung des Vantage Point Tree	22
6	Evaluation	24
6.1	Grundbegriffe	24

INHALTSVERZEICHNIS

6.2	Subjektive Bildbewertung	25
6.3	Maß der Precision	28
7	Fazit	30
	Literaturverzeichnis	31

1 Image Retrieval

1.1 Bedeutung des Image Retrieval

Ein Bild gibt uns das Gefühl, die ganze Welt in den Händen zu halten. Fotos sind Dokumente des Augenblicks, des Lebens, der Geschichte. Die Fotografie verwandelt die Welt in ein ewig fortbestehendes Angebot, aus der Wirklichkeit in das Reich der Fantasie, aus dem Schmerz in die Freude zu fliehen – durch das Fenster der Seele, das Auge. (Mario Cohen)

Bilder haben in unserer heutigen Zeit eine wachsende Bedeutung erhalten. In allen Bereichen der Multimedia gibt es eine grenzenlose Anzahl dieses Mediums. Schon allein die digitale Technik trägt einen großen Teil dazu bei. Es wird dadurch immer schwerer eine passende, computerbasierte Suche zur Auswahl einzelner Bilder zu entwickeln. Das Image Retrieval beschäftigt sich mit dem Speichern und gezielten Suchen von digitalen Bildern basierend auf Farben, Formen, Texturen und Anordnungen. Trotzdem sind die Ansätze und Lösungen meist nicht ausgereift. Das liegt nicht zuletzt an den mangelnden Evaluationen dieser Systeme.

1.2 Ausblick

In dieser Arbeit sollen drei verschiedene Ansätze zur Indexierung beim inhaltsbasierten Image Retrieval evaluiert werden. Als Basis werden dafür Farben und Formen mit Hilfe von annularen Histogrammen extrahiert. Eine Ähnlichkeitsmetrik für diese Histogramme wird über das Produkt der Eulerschen Distanz und der Histogramm Distanz gebildet, der sogenannten Distanzmetrik d_{eh} .

Unter Verwendung dieser Metrik werden folgende drei Verfahren der inhaltsbasierten N-Nearest-Neighbor Suche nach Bildern in dieser Arbeit vorgestellt:

1. Einfache, sequentielle Suche in den Histogrammen der Bilder
2. Verwendung des Evolving Tree, einer Art SOM¹, als Indexstruktur
3. Nutzung des Vantage Point Trees als Indexstruktur

¹SOM = Self-Organizing Map, ein künstliches, neuronales Netz nach Teuvo Kohonen

Abschließend sollen die drei beschriebenen Verfahren miteinander verglichen und ausgewertet werden. Im Hinblick auf das inhaltsbasierte Image Retrieval zeigen sich die Vorteile und Nachteile der Anwendung dieser Verfahren.

2 Grundlagen

2.1 RGB-Farbmodell

Es existiert ein elektromagnetisches Spektrum, das einen eingeschränkten vom Mensch sichtbaren Bereich enthält. Dieses sogenannte Farbspektrum ist in Abbildung 2.1 dargestellt. Sein Bereich verläuft zwischen einer Wellenlänge von 380nm bis 780nm. Es ist möglich durch ein Prisma, und somit durch die doppelte Brechung des Lichts, das weiße Sonnenlicht in ein kontinuierliches Spektrum der Farben im sichtbaren Bereich aufzuspalten.

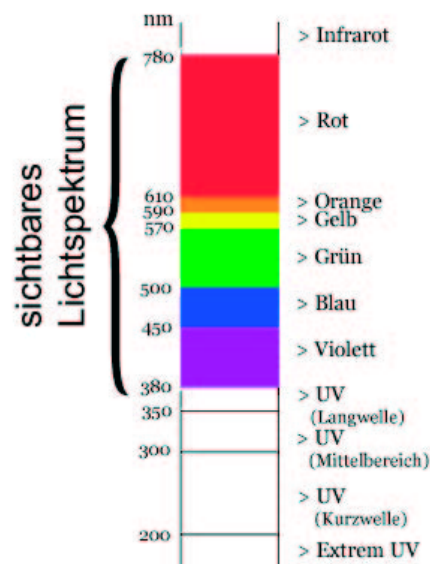


Abbildung 2.1: Das sichtbare Farbspektrum

Das RGB-Farbmodell kann einen Teil des Spektrums (2^{24} visualisierbare Farben) repräsentieren. Es stellt eine Farbe durch ihre Anteile an den drei Primärvalenzen Rot, Grün und Blau dar. In diesem Modell wird jeder Farbwert durch ein Byte, also Zahlen zwischen 0 und 255, dargestellt. Bei dem erweiterten RGB-Farbmodell steht an erster Stelle noch der Alpha-Wert eines Farbtons. Dieser Transparenz-Wert wird ebenfalls durch ein Byte dargestellt.

Das RGB-Farbmodell ist ein additives Farbmischsystem, das bedeutet, dass Weiß das Ergebnis der Farbmischung ist, wenn alle drei Farbkomponenten den Wert 255 haben. Vice

versa gilt dies für Schwarz.

Das RGB-Farbmodell wurde für das Programm ausgewählt, da es in der Computergrafik häufig verwendet wird. Das farbige Feld im CIE-Normvalenzsystem nach Rösch (Abbildung 2.2) steht symbolisch für alle für die vom menschlichen Auge sichtbaren Farben. Das additive Farbmischen hat aber nicht nur beim menschlichen Auge Bedeutung, sondern auch in der elektronischen Bildwiedergabe (PC-Monitor, Fernseher). In diesem Falle beschränkt es sich zwar auf einen kleineren Bereich, aber die Wiedergabe und die menschliche Wahrnehmung sind ähnlich. Somit können gute Ergebnisse im Bereich des Image Retrieval erzielt werden.

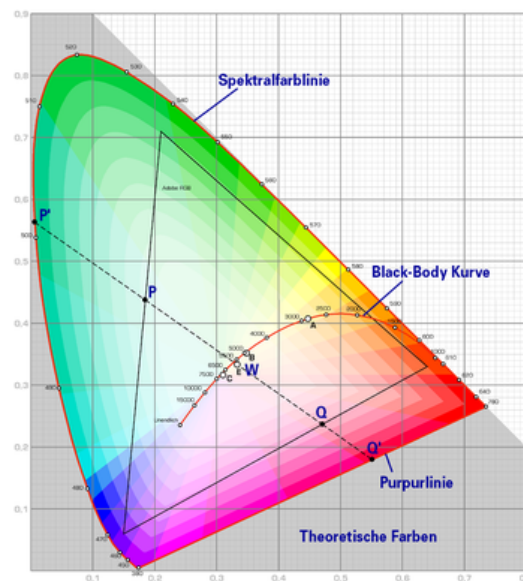


Abbildung 2.2: Die CIE-Normfarbtafel nach Rösch

2.2 Indizierte Farbtabelle

Es ist möglich, die RGB-Farbwerte als 32-Bit-Ganzzahl zu betrachten. Dem Alpha-Wert wird dabei keine Aufmerksamkeit geschenkt. Jeweils acht Bit dieser Zahl stehen für die Farben Rot, Grün und Blau. Die drei Primärvalenzen werden auf $16 \cdot 16 \cdot 4$ Farben quantisiert. In Folge dessen wird das Spektrum auf 1024 Farben reduziert. Diesen Vorgang kann man auch durch binäre Rechts-Shifts realisieren, da die Zahlen 16 und vier auch als Zweierpotenzen (2^4 , 2^2) darstellbar sind.

Blau erhält eine kleinere Auflösung, als die übrigen Farben, da das menschliche Auge schwächer auf Blau reagiert, weil die blauempfindlichen S-Zapfen seltener auftreten. Nur 12% aller Zapfen des Auges sind S-Zapfen. Dies ermöglicht eine stärkere Kompression der

Histogramme beim Image Retrieval, da für den Mensch der Blauanteil unbedeutender in Bildern erscheint und so vom Computer grobkörniger betrachtet werden kann.

Um die drei Primärvalenzen auf geringere Werte zu reduzieren, müssen sie zunächst voneinander getrennt werden. Bei Blau werden die ersten 16 Bit (dies sind die beiden anderen Farben Rot und Grün) ausgeblendet. Dieser Wert wird dann auf 2 Bit reduziert. Bei der Quantisierung von dem Farbwert Grün müssen Rot und Blau ausgeblendet werden. Anschließend wird der Wert auf 4 Bit zusammengefasst. Bei der Quantisierung von Rot wird genauso vorgegangen, nur dass hier Grün und Blau unbedeutend sind.

Das endgültige Ergebnis sind drei einzelnen Farbvariablen Rot mit vier Bit, Grün mit vier Bit und Blau mit zwei Bit. Diese Farbwerte werden dann wieder zu einer Ganzzahl verbunden. Somit erhält man eine indizierte Farbtabelle von 1024 möglichen Farbwerten.

2.3 Vorteile der indizierten Farbtabelle

Auf Grund der Quantisierung wird eine geringere Rechenzeit benötigt, da nur 2^{10} statt 2^{24} Farbwerte bestimmt werden müssen. Bei der späteren Histogrammdarstellung werden nah beieinander liegende Farbwerte als ein Farbwert quantisiert. Dadurch werden auch ähnliche Farben auf unterschiedlichen Bildern gefunden. Angesichts der Tatsache, dass Blau im Verhältnis zu den anderen beiden Farbwerten reduziert wird, rückt man mit Hilfe des Programms ein Stück näher an die optische Wahrnehmung des Menschen und blendet unwichtige Farbanteile aus.

Benutzt man viele Farben, zum Beispiel 2^{24} Farbwerte, so erzielt man zwar eine hohe Präzision, allerdings ist der Recall sehr gering. Bei der Verwendung von wenigen Farben, erreicht man einen hohen Recall, dafür aber eine geringe Präzision. Daher wurde ein vernünftiger Mittelwert von 2^{10} Farbwerten für eine gute Bildrepräsentation gefunden.

2.4 Erstellung eines Histogramms

Ein Histogramm ist in der digitalen Bildverarbeitung eine statistische Verteilung der Farbwerte eines Bildes. Es erlaubt Aussagen über die vorkommenden Grau- bzw. Farbwerte, über Kontrastumfang und Helligkeit eines Bildes abzuleiten. In einem farbigen Bild kann entweder ein Histogramm aller möglichen Farben oder drei Histogramme der einzelnen Farbkanäle erstellt werden.

Ein Beispiel für ein Histogramm eines Bildes über alle Farben, kann man in der Abbildung 2.3 betrachten.

Zur Erstellung eines Farbhistogramms zu einem Bild sind folgende Schritte notwendig. Zuerst wird die Farbanzahl des Bildes festgelegt. Somit wird ein Farbmodell ausgewählt und wieviele Bits jeder Farbe zugeordnet werden. In diesem Fall wurde dies durch die indizierte Farbtabelle entschieden. Danach werden die absoluten Häufigkeiten jeder Farbe berechnet,

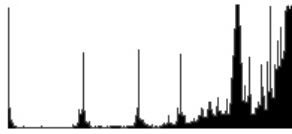


Abbildung 2.3: Ein Histogramm eines Farbbildes

dass heißt, es werden alle Pixel gezählt, die von einem bestimmten Farbton kommen. Alle Bildpunkte eines Farbwertes werden dann in sogenannte Bins („Eimern“) abgelegt. Anschließend werden die relativen Häufigkeiten jeder Farbe bestimmt:

$$\frac{\text{AnzahlPixel pro Farbton}}{\text{Anzahl aller Pixel}}$$

In Folge dessen erhält man ein definiertes Histogramm der Farbverteilung eines Bildes in Abhängigkeit der verwendeten Farbtabelle.

2.4.1 Angulare und hybride Histogramme

Ein angulares Histogramm stellt die Verteilung der Farbpunkte eines Bildes durch vorgegebene Kreissektoren, wie in Abbildung 2.4 deutlich wird. Hierbei wird ein Bin-Mittelpunkt und die Anzahl der Sektoren festgelegt. Danach werden die statistischen Häufigkeiten der Punktansammlungen jedes Kreissektors für jede einzelne Farbe innerhalb des Bildes ermittelt.

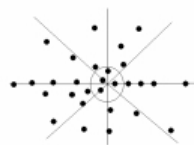


Abbildung 2.4: Ein angulares Histogramm

Ein hybrides Histogramm repräsentiert die Verteilung der Farbpunkte durch Kreissektoren sowie zusätzlich durch eine gewählte Anzahl von Ringen mit gleichen Abständen voneinander (siehe Abbildung 2.5). Zunächst wird ein Bin-Mittelpunkt und die Anzahl der Winkel und Kreise festgelegt. Anschließend ergeben sich die statistischen Häufigkeiten innerhalb der Schnittflächen der Kreissektoren und Ringe für die einzelnen Farben der Farbtabelle.

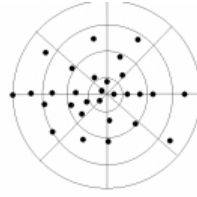


Abbildung 2.5: Ein hybrides Histogramm

2.4.2 Annulare Histogramme

Ein annulares (ringförmige) Histogramm (Beispiel siehe Abbildung 2.6) entsteht durch Ringe mit gleichen Abständen voneinander, die auf einen Bereich des Bildes vom Bin-Mittelpunkt zu den Rändern des Bereichs gelegt werden. Für jede einzelne Farbe innerhalb des Bildes werden wiederholt unter Nutzung der indizierten Farbtabelle die statistischen Häufigkeiten des Farbvorkommens innerhalb eines jeden Ringabschnitts ermittelt.

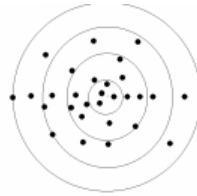


Abbildung 2.6: Ein annulares Histogramm

In diesem Zusammenhang wurde zunächst eine Auswahl der Farbgegend, des sogenannten Bins, vorgenommen. Es erfolgte eine Zuordnung der Farbe in die indizierte Farbtabelle. Anschließend folgt eine Ermittlung des Bin-Mittelpunktes (x_q, y_q) und des maximalen Radius (r_q) vom Mittelpunkt zum äußersten Punkt des Bins.

$$x_q = \frac{1}{|S_q|} \sum_{x,y \in S_q} x$$

$$y_q = \frac{1}{|S_q|} \sum_{x,y \in S_q} y$$

$$r_q = \max(\sqrt{(x - x_q)^2 + (y - y_q)^2})$$

Der Maximalradius wird in 16 Teilradien und dadurch in 16 Ringe unterteilt. Anschließend wird die relative Häufigkeit aller Farbpunkte eines Bins innerhalb eines Kreisabschnitts ermittelt. Die relativen Häufigkeiten der einzelnen Farben eines Rings werden in einer Matrix gespeichert.

$$H(n,m) = \begin{bmatrix} h_1 1 & \cdots & 0,01 & \cdots & h_m 1 \\ \vdots & \ddots & \vdots & & \vdots \\ h_1 n & \cdots & 0,05 & \cdots & h_m n \end{bmatrix}$$

n ...Anzahl der Ringe, m ...Anzahl der Farben

2.4.3 Nutzung des annularen Histogramms

Die Vorteile des angularen und des hybriden Histogramms sind die Betrachtung der gleichen Perspektiven, Positionen und Richtungen von Bilddetails. Andere gleichfarbige Elemente stören die Suche nicht, da die genaue Position (angular und hybrid) des Elements festgelegt ist und die geometrische Form (hybrid) definiert wird. Die Nachteile sind die Festlegung der Position eines Elementes und beim hybriden Histogramm sogar die Festlegung der genaueren Position der Farbbereiche. Das angular Histogramm betrachtet außerdem keinerlei geometrische Formen.

Das annulare Histogramm hingegen hat mehr Vorteile, die für das farbbasierte Image Retrieval notwendig sind. Es werden relative Häufigkeiten des Farbvorkommens genutzt. Dadurch funktioniert die Suche bei kleineren oder größeren Auflösungen problemlos, da die relative Verteilung der Farbpunkte gleich bleibt. Aufgrund der Betrachtung der Ansammlung von Bildpunkten innerhalb eines Rings ist die geometrische Präzision und somit die Form eines Objekts gewährleistet. Gleichzeitig gibt es keine Positions- oder Richtungsabhängigkeiten der Bilddetails, da die relative Farbverteilung gleich bleibt. In dem Vergleich von [AR99] zeigte das annulare Histogramm eine bessere Performanz im Vergleich zum angularen und hybriden Pendant.

3 Distanzklassifikation

3.1 Bildklassifikation

Um eine Klassifikation zu ermöglichen, werden die notwendigen Daten sequentiell gespeichert. Zuerst wird die Anzahl der Farben, die Anzahl der Ringe des Histogramms und die Anzahl der Bilder gesichert. Dies ist notwendig, um bei der Vergleichsklassifikation dasselbe annulare Histogramm zu erstellen, wie es in der Trainingsmenge genutzt wurde. Jedes neue Bild, für das ähnliche Bilder gefunden werden sollen, erhält ebenfalls eine Histogramm-Matrix q auf Basis der vorgegebenen Anzahl Farben und Ringe. Für jedes Bild innerhalb der Trainingsmenge wird eine Histogramm-Matrix h empfangen. Diese Matrizen und die zu den Trainingsbildern zugehörigen relativen Pfade werden ebenfalls akkumuliert. Die Matrix des Querybildes wird mit den Matrizen aus der Datei sequentiell verglichen.

3.1.1 Euler Distanz

Durch diese Distanzformel fallen große Abweichungen bei großen relativen Häufigkeitswerten nicht so sehr ins Gewicht, wie große Abweichungen bei kleinen Häufigkeiten.

$$Euler = \sqrt{\sum_{i,j} \frac{(q(i,j) - h(i,j))^2}{q(i,j) + h(i,j)}}$$

q ...Beispiel-Histogramm, h ...Datafile-Histogramm

Der Zähler dieser Formel beinhaltet die Euklidische Distanz. In dieser Abstandsformel gibt es nur einen Vergleich zwischen den identischen Bins der jeweiligen Histogramme. Zwei unterschiedliche Bins können wahrnehmbar ähnliche Farben darstellen, aber sie werden nicht kreuzweise verglichen. Alle Bins tragen gleichmäßig dem Abstand bei. Durch die Quadrierung werden größere Abweichungen stärker gewichtet. Damit diese Distanz nicht zu einseitig betrachtet wird, ist es notwendig, sie mit der Summe der beiden Farbhäufigkeiten von q und h ins Verhältnis zu setzen.

Je kleiner das Ergebnis ist, umso genauer ist die Übereinstimmung.

3.1.2 Histogramm Distanz

Diese Abstandsgleichung soll verdeutlichen, zu wieviel Prozent das Datafile-Histogramm h mit dem Query-Histogramm q übereinstimmt.

$$Histogram = \frac{\max(q(i, j), h(i, j))}{\min(q(i, j), h(i, j))} - 1$$

q ...Beispiel-Histogramm, h ...Datafile-Histogramm

Der Zähler stellt die Summe aus dem Maximum der einzelnen Elemente der Histogramme dar. Im Nenner wird die Summe aller minimalen Elemente des Beispiel-Histogramms und Datafile-Histogramms berechnet. Damit die Distanz zwischen Null und Eins liegt muss Eins von dem Quotienten abgezogen werden. Je größer das Ergebnis ist, umso genauer ist die Übereinstimmung der beiden Histogramme.

3.1.3 Distanzmetrik

Es ergibt unterschiedliche und oft ungenaue Ergebnisse, wenn nur nach einer der beiden Distanzkriterien die Bilder sortiert werden. Bei manchen Treffern ist die Distanz bei der Euler-Distanz und der Histogramm-Distanz groß, aber optisch gesehen erscheint das gefundene Bild dem Anfragebild nicht ähnlich zu sein. Es gibt auch die Fälle, das beide Distanzkriterien einen niedrigen Wert haben und das gefundene Bild ebenfalls nicht zu passen scheint.

Deshalb wird das Produkt der beiden Klassifikationen gewählt (wie in Algorithmus 1 dargestellt):

$$d_{eh} = Euler * Histogram$$

Diese Ergebnisse aufsteigend sortiert, ergeben eine gute Auswahl an Trefferbildern, die auch optisch dem Anfragebild ähneln.

Der Nachteil des Programms ist die hohe Rechenzeit, die durch die sequentielle Suche verursacht wird. Je größer die Auflösungen der Bilder sind, umso länger dauert die Trainings- und Suchzeit. Hierfür wäre eine andere, geschickte Indexierung die Lösung.

Algorithmus 1 Die Distanzklassifikation

INPUT: Sei q die Querymatrix und h eine der gespeicherten Histogramm-Matrizen, sei weiterhin n die Anzahl der Ringe des annularen Histogramms und m die Anzahl der Farben

OUTPUT: d_{eh} die Distanzmetrik

for $i = 0$ to n **do**

for $j = 0$ to m **do**

$Euler = \sqrt{\sum_{i,j} \frac{(q(i,j) - h(i,j))^2}{q(i,j) + h(i,j)}}$ {Berechne die Euler Distanz}

$Histogram = \frac{\max(q(i,j), h(i,j))}{\min(q(i,j), h(i,j))} - 1$ {Berechne die Histogramm Distanz}

end for

end for

$d_{eh} = Euler * Histogram$ {Berechne das Produkt von Euler Distanz und Histogramm Distanz}

4 Indexierung mit Hilfe des Evolving Tree

4.1 Probleme bei üblichen neuronalen Netzen und Self-Organizing Maps

Die von Teuvo Kohonen entwickelte Self-Organizing Map, oder Kohonennetz genannt, ist ein künstliches neuronales Netz mit multidimensionalen Eingängen und einer planaren topologischen Struktur im Inneren siehe Abbildung 4.1.

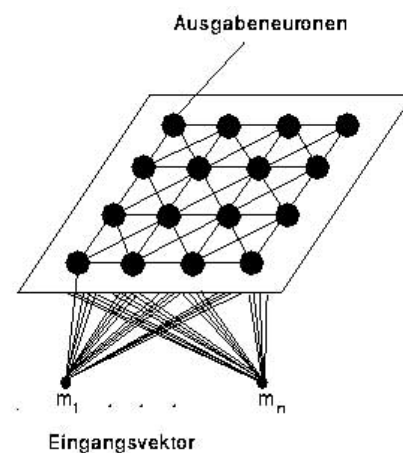


Abbildung 4.1: Die Struktur einer Kohonen Map

Es wird häufig als Datenanalysewerkzeug für Aufgaben im Bereich der Datenvisualisierung und Clusterung benutzt. Allerdings ist es nachteilig, dass die Größe des Netzes nicht automatisch ermittelbar ist. Daher müssen einige Experimente im Voraus durchgeführt werden, um eine möglichst optimale Größe der Abbildung zu finden. Aufgrunddessen ist das Trainieren und Nutzen dieses Netzes viel zu zeitaufwendig.

In [Pak03] wurde ermittelt, dass ein System benötigt wird, welches ein intuitives Gridwachstum und eine hierarchische Suche innerhalb der Grids unterstützt.

4.2 Definition des Evolving Trees

Der Evolving Tree ist ein neuartiges, neuronales Netz in der Form eines Baumes, entwickelt vom Laboratory of Computer and Information Science der Helsinki University of Technology.

Dieser Baum unterstützt Gridwachstum und die hierarchische Suche innerhalb der Grids. Jeder Knoten des Baumes enthält:

1. einen Prototypenvektor v ,
2. den Index i des Knotens,
3. einen Wichtungsvektor w_i , sowie
4. einen Zähler b_i , welcher zählt, wie oft ein Knoten in der Trainingsphase am besten zum Eingangsvektor gepasst hat.

4.3 Evolving Tree Algorithmus

Zunächst wird ein einzelner Knoten erzeugt. Dieser erhält eine vordefinierte Menge an Kindknoten. Die Wichtungsvektoren der Kinder bekommen den gleichen Wert des Wichtungsvektors des Elternknotens. Dann wird die Best Matching Unit (BMU) zwischen den Kindern gewählt und die Wichtungsvektoren dementsprechend angepasst. Dieser Vorgang wird später noch etwas genauer definiert.

Durch diese zufällige Generierung zu Beginn werden die Kindknoten unterschiedlich gewichtet. Es entsteht eine Baumstruktur mit einer bestimmten Menge an Blattknoten und einem Wurzelknoten. Alle künftigen Operationen werden nur auf den Blattknoten ausgeführt. Einmal erstellt, wird der Wurzelknoten nicht mehr geteilt.

Wenn ein Blattknoten im Verlauf einen sogenannten „Splitting Threshold δ “ erreicht, wird er wieder geteilt und so weiter. Somit optimiert sich der Baum rekursiv während der Trainingsphase.

Die BMU wird in einem Top-Down-Prozess gesucht, wie in Abbildung 4.2 dargestellt. Man beginnt am Wurzelknoten. Dann werden die Kindknoten untersucht. Es wird ein Knoten gesucht, dessen Prototypvektor nach einer vorgegebenen Distanzmetrik am nächsten an dem Trainingsvektor liegt. Wenn dieser Knoten ein Blattknoten ist, dann wurde die BMU gefunden. Falls dies nicht der Fall ist, werden seine Kindknoten wiederum untersucht und der passende solange gesucht, bis ein Blattknoten erreicht wird. Dieses Suchen der BMU innerhalb des Baumes wird mit Hilfe des in Algorithmus 2 dargestellten Pseudokodes realisiert.

Demnach funktionieren die Baumknoten des Evolving Tree als hierarchischer Suchbaum für die Blattknoten.

Nach der Ermittlung der BMU werden der Wichtungsvektor der BMU sowie die Wichtungsvektoren seiner Nachbarn gegenüber dem Trainingsvektor aktualisiert:

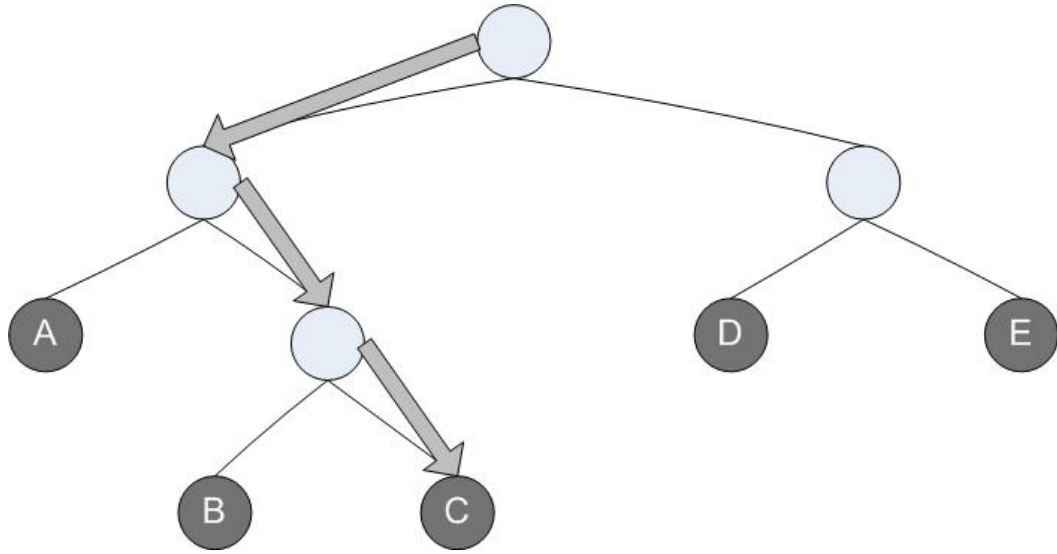


Abbildung 4.2: Algorithmus zur Auffindung der BMU

$$w_i(t+1) = w_i(t) + h_{ci}(t)[x(t) - w_i(t)]$$

$x(t)$ ist der Trainingsvektor, c der Index der BMU und $h_{ci}(t)$ stellt die Nachbarschaftsfunktion dar.

Zusätzlich wird der Zähler bc der BMU inkrementiert:

$$bc(t+1) = bc(t) + 1$$

Wenn nach der Inkrementierung $bc(t+1) = \delta$ sein sollte, dann wird die BMU wieder geteilt.

Dies hat somit den Wachstum des Baums zur Folge. Die Geschwindigkeit des Wachstums wird durch den Parameter δ kontrolliert.

Die Nachbarschaftsfunktion wird durch die Distanzmetrik d_{eh} aus Abschnitt Abschnitt ?? realisiert.

Innerhalb eines Kohonennetzes entspricht der Wert der Distanz dem Gridabstand zwischen dem zu aktualisierenden Knoten und der BMU. Dieser ist leicht zu ermitteln, da ein Kohonennetz symmetrisch und statisch ist, sowie eine reguläre Struktur besitzt. Keine dieser Eigenschaften trifft auf den Evolving Tree zu.

Es gibt eine relativ einfache Möglichkeit, die äquivalente Distanz innerhalb des Baumes zu berechnen. In einer Baumstruktur ist immer der geringste Abstand zwischen zwei Knoten signifikant.

Algorithmus 2 Das Auffinden der BMU**FUNKTION:** FindeBMU(*datenvektoren*, *knoten*)**INPUT:** *datenvektoren* ein Container für die trainierten Datenvektoren, *knoten* ein Knoten des Baumes (zu Beginn die Wurzel)**OUTPUT:** *BMU* die Best Matching Unit**if** *knoten* ist kein Blattknoten **then** Teste, welches der Kinder von *knoten* die kleinste Distanz hat in Bezug auf den zugehörigen Datenvektor aus *datenvektoren* **if** *knoten.linksKind* hat kleinste Distanz d_{eh} **then** FindeBMU(*datenvektoren*, *knoten.linksKind*) **else** FindeBMU(*datenvektoren*, *knoten.rechtesKind*) **end if****end if***BMU* = Kind mit kleinster Distanz**return** *BMU*

Es wird einfach die Anzahl der Sprünge von Knoten zu Knoten ermittelt, die ausgeführt werden müssen, um von der BMU zum gewünschten Knoten entlang des kürzesten Pfades zu gelangen (Abbildung 4.3). Die Distanz ist dann die Anzahl der Sprünge minus Eins. Eins muss subtrahiert werden, da nur Distanzen zwischen den Blattknoten betrachtet werden.

Beispiel: Die nächst möglichen Blattknoten haben einen gemeinsamen Elternknoten. Bei Betrachtung von Abbildung 4.3, wären das zum Beispiel die Knoten *B* und *C*. Um von einem zum anderen Blatt zu gelangen, benötigt man zwei Sprünge. Diese „engsten“ Nachbarn sollen eine Distanz von Eins haben, deshalb wird immer eine Eins subtrahiert, um die Konsistenz aller Distanzen beizubehalten.

Aufgrund des Aufbaus eines Evolving Trees wird garantiert, dass nur ein einziger Pfad zwischen jeweils zwei Blattknoten existiert. Deshalb ist diese Distanzfunktion eindeutig und symmetrisch. Dadurch, dass der Pfad symmetrisch ist, ist dieser wiederum der kürzeste. Dies zeigt, dass die Funktion, mathematisch sinnvoll ist.

Weil die Teilbäume eines Baumes zu verschiedenen großen Bereichen des Datenraumes wachsen, sind die Distanzen innerhalb eines Teilbaums kürzer als zwischen zwei verschiedenen Teilbäumen.

4.4 Evolving Tree als Index

Der Evolving Tree kann auch als baum-strukturierter Index in einem n -dimensionalen metrischen Raum gesehen werden. Jeder Knoten erhält die Werte des relativen Pfades eines Bildes und seiner Histogramm-Matrix. Der Baum wird wie vorher beschrieben aufgebaut,

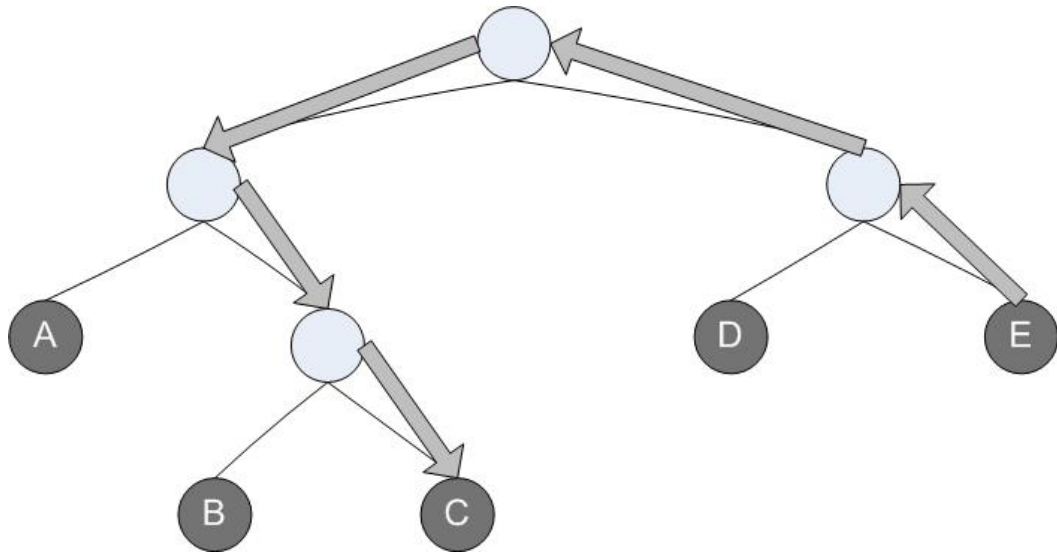


Abbildung 4.3: Ein Beispiel zur Distanzermittlung

trainiert und weiterentwickelt. Mit Hilfe der vorgeschlagenen Distanzmetrik d_{eh} werden die am besten passenden Knoten gesucht. Es muss jetzt beachtet werden, dass nicht nur ein Knoten (BMU) das gewünschte Ergebnis ist, sondern eine Menge von mehreren ähnlichen Bildern gesucht wird. Dies kann durch eine N-Nearest-Neighbor Suche von der BMU ausgehend realisiert werden. Diese Art der Suche wird im folgenden Kapitel näher definiert. Die Umsetzung des Suchalgorithmus für den Evolving Tree kann in Algorithmus 3 betrachtet werden.

Algorithmus 3 Die N-Nearest-Neighbor Suche im Evolving Tree

FUNKTION: *Suche(anfragebild, n)***INPUT:** *anfragebild* die Daten des Beispielbildes, *n* die Anzahl der *n* nächsten Nachbarn**OUTPUT:** *nearestNeighbors* eine distanzsortierte Liste der gefundenen Knoten als nächste Nachbarn zum Anfragebild*stack* \leftarrow *push(stack, WurzelknotendesE – Tree)***while** nicht leer(*stack*) **do** *knoten* \leftarrow *pop(stack)* **if** *knoten* ist kein Blattknoten **then** Nimm das Kind *kind* von *knoten*, dass noch nicht besucht wurde **if** *kind* gefunden **then** *push(stack, kind)* **else** *pop(stack)* **end if** **else** Füge *knoten* am Ende der Liste *nearestNeighbors* ein **end if** Markiere *knoten* als besucht**end while****return** *nearestNeighbors*

5 Vantage Point Tree als Indexstruktur

5.1 Definition des Vantage Point Tree

Der Vantage Point Tree, oder auch VP-Baum genannt, ist ein BSP-Baum (Binary Space Partitioning-Baum) der Daten im metrischen Raum mit Hilfe von abnehmenden, metrischen Werten isoliert. Der VP-Baum wird hauptsächlich im Zusammenhang mit der N-Nearest-Neighbor Suche (kurz: NNS) genutzt.

Bei dieser Suchmethode wird ein Objekt durch die Mehrheitswahl seiner Nachbarn klassifiziert. Dabei ist das Objekt als der Klasse am meisten Gemeinsames inmitten von k nächsten Nachbarn festgelegt.

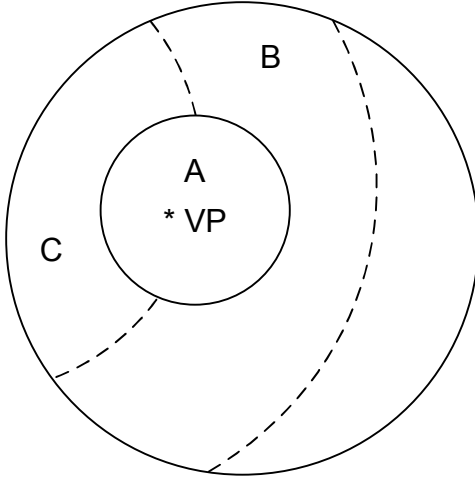
Die Nachbarn werden von einer Menge von Objekten entnommen, für welche die korrekte Klassifikation bekannt ist. Zur Identifikation der Nachbarn werden die Objekte von einem Positionsvektor in einem multidimensionalen Merkmalsraum repräsentiert. Diese werden mit einer Abstandsfunktion auf ihre Entfernung zueinander abgeschätzt.

Als Merkmale des Bildes werden k -dimensionale annulare Histogramme genutzt. Auch hier wird die in Abschnitt ?? vorgeschlagene Distanzmetrik d_{eh} verwendet und die inhaltsbasierte Bildsuche als multidimensionales N-Nearest-Neighbor Suchproblem definiert. Für die N-Nearest Neighbor Suche ist es notwendig, dass die Suchstruktur Distanzinformationen zwischen Punkten innerhalb einer Partition und der Partitionsgrenzen enthält. Der VP-Baum teilt den Suchraum mit Hilfe von relativen Distanzen zwischen dem Vantage Point und seinen Kindern. Aus diesem Grund ergibt sich diese aus der Distanz zwischen einem Punkt und den Grenzen der Partition, zu der er gehört. Der VP-Baum setzt Grenz-sphären (Kugeln) ein und deshalb sollte die Genauigkeit der Raumaufteilung besser sein, als wenn ein Würfel zur Begrenzung genutzt wird. Denn wenn die Raumdimension wächst, ist das Wachstum des Kugelvolumens im Vergleich zum Volumen eines Würfels geringer, von dem es umschrieben werden kann. In Abbildung 5.1 wird dieser Sachverhalt deutlich.

5.2 Algorithmus des Vantage Point Tree

5.2.1 N-Nearest-Neighbor im VP-Baum

Sei ein metrischer Raum (S, d) gegeben und seine endliche Teilmenge $S_D < S$ repräsentiert eine Datenbasis. Es wird ein Datenpunkt v aus S_D als Vantage Point gewählt. Für jeden anderen Punkt $p \in S_D - \{v\}$ werden die Distanzen zu v berechnet und der Mittelwert μ wird bestimmt.



A, B, C sind Teilmengen des metrischen Raums
(Knoten des Baumes)

VP ist der Vantage Point

Abbildung 5.1: Beispiel der Grenzsphären des VP-Baums

Binär betrachtet, stellt sich das Problem folgendermaßen: Die gesamte Datenmenge S_D wird in Übereinstimmung zum Vantage Point und dem Mittelwert in zwei Teile geteilt. Sei S_{\leq} eine Menge von Punkten, deren Distanzen von v kleiner oder gleich dem Mittelwert sind und $S_{>}$ eine Menge von Punkten, deren Abstände größer als μ sind.

Angenommen der nächste Nachbar für einen Punkt q wird gesucht, dessen Distanzen von q kleiner als ein bestimmter Schwellwert δ sind. Es stellt sich heraus, dass für $d(v, q) \leq \mu - \delta$ nur die Teilmenge S_{\leq} betrachtet werden muss und für $d(v, q) > \mu + \delta$ nur die Teilmenge $S_{>}$.

Wie in [Mar07] erklärt, basiert diese Beobachtung auf der Dreiecksungleichung:

Wenn $d(v, q) \leq \mu - \delta$ ist, dann gilt für jedes $p \in S_{>}$

$$d(q, p) \geq |d(v, p) - d(v, q)| > |\mu - (\mu - \delta)| = \delta$$

also ist $d(q, p) > \delta$. Die Submenge kann demnach ausgeschlossen werden.

Effektiverweise kann eine Hälfte des Suchraums abgeschnitten werden, wenn folgendes gilt: die Potenz von der Menge $S_{>}$ ist annähernd gleich S_{\leq} und $d(v, q)$ trifft nicht auf folgende, zweiseitige Ungleichung zu:

$$\mu - \delta < d(v, q) \leq \mu + \delta$$

Allgemein formuliert: An jedem Knoten des VP-Baums, begrenzt der Vantage Point m die Submengen S_i durch die Distanz μ_i . Wenn die Hypersphäre nur in eine Teilmenge S_i fällt, braucht der Algorithmus auch nur diese eine Teilmenge betrachten. Ansonsten müssen mehrere Teilmengen (Teilbäume) erkundet werden. Je kleiner der Schwellwert δ gesetzt wird,

umso weniger Teilbäume müssen erforscht werden, allerdings steigt damit auch das Risiko, dass kein Nearest Neighbor gefunden wird. Wodurch die Suche mit einem größeren Wert für δ wiederholt werden muss. Deshalb ist der Initialwert für δ unendlich und wird dynamisch verbessert. Während der Traversierung wird δ an den Wert der Distanz zwischen dem Anfrageobjekt und dem aktuellen n-ten nächsten Nachbarn angepasst. Dadurch garantiert δ das der n-te nächste Nachbar maximal die Distanz von δ zum Anfragepunkt besitzt. Deshalb wird δ durch die Distanzen der aktuellen Datenobjekte der Blattknoten, die während der Tiefensuche aufeinanderstoßen, begrenzt. Dieser Suchalgorithmus für den VP-Baum ist zum genaueren Verständnis noch als Pseudocode in Algorithmus 4 angegeben. Die Methode der N-Nearest-Neighbor Suche hat eine sehr gute Performanz, wie [AWcF00] zeigt.

5.2.2 Auswahl des Vantage Point

Jeder Knoten des Baums wird eindeutig einem Vantage Point zugeordnet. Wenn die Suchoperation mehrfache Abzweigungen traversiert, muss auch eine Abstandsberechnung an der Wurzel jedes Abzweiges erfolgen. Wenn also nur ein Vantage Point zur Regionsteilung mit den Knoten derselben Ebene assoziiert wird, dann ist nur eine Abstandsberechnung an jeder Nicht-Blatt Ebene notwendig.

An der Wurzel wird der erste Vantage Point gewählt. Dann wird der zweite Vantage Point für die nächste Ebene so gewählt, dass er derjenige ist, der am weitesten vom ersten Vantage Point entfernt ist. Der Dritte ist wieder am weitesten entfernt von den beiden ersten, usw. Dadurch wird eine effektive Teilung der Datenmenge gefördert.

Aufgrunddessen, dass für jede Ebene nur ein Vantage Point existiert, ist die Anzahl der Abstandsberechnungen an den Nicht-Blattknoten gleich der Anzahl der Nicht-Blattknoten Ebenen des Baums.

Dadurch wird die Baumgröße kleiner, die Verzweigungen an Nicht-Blättern größer und somit natürlich auch die Performanz besser.

Das Grundprinzip des Algorithmus zur Auffindung des Vantage Point ist nocheinmal im Pseudocode Algorithmus 5 verdeutlicht.

5.2.3 Persistenzproblem

Beim VP-Baum ist es schwierig Updates durchzuführen und eine globale Reorganisation der Struktur könnte das Resultat sein. Ein Knoten kann nicht einfach in geeigneter Weise geteilt werden, wobei diese, wie im B-Baum, nach in Richtung Wurzel propagiert wird.

Bei der Teilung eines Elternknotens, müssen auch dessen Kinder geteilt werden. Deshalb muss die Zerteilung abwärts fortgeführt werden. Ähnliche Probleme können beim Löschen eines Knotens auftreten, wenn dadurch andere Knoten wiederum verbunden werden müssen. Andererseits kann das Einfügen oder Löschen eines Elementes innerhalb eines Knotens auch wiederum notwendige Veränderungen im Elternknoten hervorrufen. Dies würde dann dem Bottom-Up-Prinzip entsprechen.

Algorithmus 4 Die N-Nearest-Neighbor Suche im Vantage Point Tree

FUNKTION: $\text{Suche}(q, n, \text{knoten}, \text{delta}, \text{nearestNeighbors})$ **INPUT:** Das Anfrageobjekt q , die Anzahl der n nächsten Nachbarn n , ein Knoten des Baums knoten (zu Beginn die Wurzel), delta als Distanzparameter (zu Beginn auf Unendlich), nearestNeighbors die Menge der nächsten Nachbarn (zu Beginn leer)**OUTPUT:** die aktualisierten Werte von nearestNeighbors und delta

```
if  $\text{knoten}$  ist Blatt then
  for all  $v \in \text{knoten}$  do
    if  $d_{eh}(v, q) \leq \text{delta}$  then
      Füge  $v$  in  $\text{nearestNeighbors}$  ein (in aufsteigender Reihenfolge)
    end if
    if  $|\text{nearestNeighbors}| = n$  then
       $\text{delta} = d_{eh}(\text{nearestNeighbors}_n, q)$ 
    end if
  end for
else
   $\text{dist} = d_{eh}(v, q)$ 
  if  $\text{dist} < \text{knoten.mittelwert}$  then
    if  $\text{dist} < \text{knoten.mittelwert} + \text{delta}$  then
       $\text{Suche}(q, n, \text{knoten.linksKind}, \text{delta}, \text{nearestNeighbors})$ 
    end if
    if  $\text{dist} \geq \text{knoten.mittelwert} - \text{delta}$  then
       $\text{Suche}(q, n, \text{knoten.rechtesKind}, \text{delta}, \text{nearestNeighbors})$ 
    end if
  else
    if  $\text{dist} \geq \text{knoten.mittelwert} - \text{delta}$  then
       $\text{Suche}(q, n, \text{knoten.rechtesKind}, \text{delta}, \text{nearestNeighbors})$ 
    end if
    if  $\text{dist} < \text{knoten.mittelwert} + \text{delta}$  then
       $\text{Suche}(q, n, \text{knoten.linksKind}, \text{delta}, \text{nearestNeighbors})$ 
    end if
  end if
end if
end if
```

Algorithmus 5 Die Auswahl des Vantage Point

INPUT: Eine zufällige Menge an Vantage Points *vantagePoints* aus der gesamten Datenmenge

OUTPUT: *vantagePoint*

for all *vantagePoint* \in *vantagePoints* **do**

 Berechne die Distanz des *vantagePoint* zu jedem einzelnen Objekt aus der Datenmenge

end for

Berechne den Mittelwert und die Standardabweichung aller Distanzen

return Den *vantagePoint*, der die größte Standardabweichung hat

Aus diesem Grund basiert der Aufbau des VP-Baums auf einem B-Baum, denn für diese Struktur wurden schon Lösungen für diese Probleme gefunden. Der maximale Verzweigungsfaktor von Nicht-Blättern und die maximale Anzahl von Objekten, beinhaltet in einem Blatt, sind begrenzt durch die Seitengröße (4 KB) wie in [AWcF00] herausgefunden.

Für ein Nicht-Blatt und eine Nicht-Wurzel variiert die Anzahl der Kindknoten zwischen einem Minimalwert und einem Maximalwert m , wie bei einem B-Baum. Das Minimum sollte nicht größer als $\lceil \frac{m}{2} \rceil$ und m sollte mindestens zwei sein.

5.3 Umsetzung des Vantage Point Tree

Im Projekt wurde ein B+-Baum als Grunddatenstruktur gewählt. Ein B+-Baum (siehe Beispiel in Abbildung 5.2) ist eine Erweiterung des B-Baums, wobei die eigentlichen Daten nur in den Blattknoten stehen und die restlichen Knoten Schlüssel enthalten. Das bedeutet, dass in den einzelnen Elementen eines Blattknotens die Histogramm-Matrix und der relative Pfad eines Bildes gespeichert werden. Die Schlüssel der Nicht-Blattknoten sind somit die Mittelwerte μ_i der Distanzen.

Per Zufall wird aus der gegebenen Datenmenge von Bildern mit ihren Histogrammen ein Vantage Point gewählt. Dann werden zu allen anderen Bildern der Datenmenge die Abstände vom Vantage Point mit Hilfe der Distanzmetrik d_{eh} berechnet. Diese Bilder werden dann anhand der Distanzen absteigend sortiert. Da die einzelnen Bilder soweit wie möglich voneinander entfernt sein sollen, um eine sinnvolle Strukturierung des B+-Baumes zu erzeugen. In die Nicht-Blätter werden nun die Mittelwerte aus den Distanzen der Bilder des jeweiligen Blattkindknotens gespeichert.

Wenn nun ein Querybild gegeben ist und die ähnlichsten Bilder gefunden werden sollen, dann muss mit Hilfe der N-Nearest-Neighbor Suche und der Distanzmetrik d_{eh} die passenden Bilder innerhalb des Baums ermittelt.

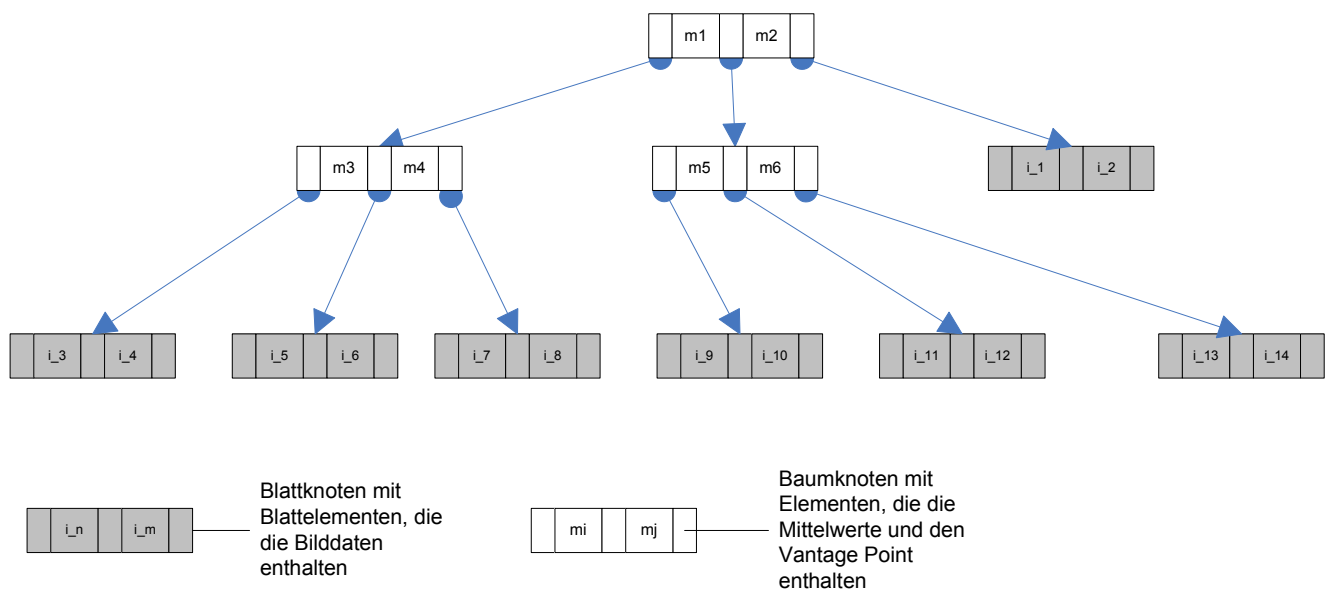


Abbildung 5.2: Beispiel eines B+-Baumes

6 Evaluation

6.1 Grundbegriffe

Evaluation (Wortstamm lat.: „valuere“= bewerten) bedeutet im Zusammenhang mit dem Image Retrieval die Beschreibung, Analyse und Bewertung von Prozessen anhand von festgelegten Kriterien. Die Prozesse sind dabei die drei unterschiedlichen Indexierungsverfahren, die in dieser Arbeit bereits beschrieben und analysiert wurden. Die festgelegten Kriterien sind die bestmögliche Rückgabe ähnlicher Bilder aus subjektiver Sicht und mit Hilfe der Precisionberechnung.

Der Recall und die Precision sind Maße zur Beurteilung der Güte des Ergebnisses einer Recherche. Der Recall (deutsch: „Abruf, Treffer“) ist die Wahrscheinlichkeit, mit der ein relevantes Bild gefunden wird. Er beschreibt die Vollständigkeit eines Suchergebnisses. Die Formel für diese Wahrscheinlichkeitsberechnung lautet:

$$Recall = \frac{|R \cap P|}{|R|}$$

Die Anzahl der Elemente aus der Vereinigung aller relevanten, positiv gefundenen Bilder steht im Verhältnis zur Anzahl aller relevanten Bilder.

Die Precision (deutsch: „Genauigkeit“) definiert die Genauigkeit eines Suchergebnisses. Sie ist die Wahrscheinlichkeit, mit der ein gefundenes Bild relevant ist. Die Formel dieser Wahrscheinlichkeitsberechnung ist:

$$Precision = \frac{|R \cap P|}{|P|}$$

Die Anzahl der Elemente aus der Vereinigung aller relevanten, positiv gefundenen Bilder steht im Verhältnis zur Anzahl aller positiv gefundenen Bilder.

Die Anzahl gefundener Objekte wird auf eine Menge von 20 Bildern beschränkt. Es wird davon ausgegangen, dass alle gefundenen Bilder für jedes der drei Indexierungsverfahren positiv, gefundene Objekte sein können ($|P|$ ist immer 20). Allerdings ändert sich die Kardinalität der Vereinigungsmenge $|R \cap P|$ beim VP-Baum und beim E-Baum, da P hier meistens nicht alle Bilder aus R enthält (bei der sequentiellen Suche gilt $P = R$). Aus diesem Grund kann der Recall in diesem Fall auch nicht bestimmt werden, da das Suchergebnis immer vollständig sein wird. Allerdings kann man die Genauigkeit der gefundenen Bilder berechnen, weil die Anzahl der relevanten, positiv gefundenen Bilder variieren kann.

Die Ergebnisse der sequentiellen Suche gelten als Goldstandard, da diese auf jeden Fall den vollständigen Suchraum untersucht und somit die besten Ergebnisse liefert. Precision und Recall sind dabei immer Eins, da hier $R = P$ gilt.

6.2 Subjektive Bildbewertung

Die subjektive Bildbewertung ist eine sehr einseitige Betrachtung. Allerdings ist sie für das Image Retrieval eine bedeutende Analyse. Es sollen dem Anschein nach Bilder gefunden werden, die auch ein Mensch ausgewählt haben könnte. Die Schnittstelle Computer soll nicht bemerkt werden. Bei dieser Art der Bewertung werden immer nur fünf Ergebnisbilder der jeweiligen Verfahren im Dokument dargestellt, um eine Vorstellung der Resultate zu ermöglichen.

Auf den ersten Blick wirken einige Ergebnisse des Evolving Tree exakter, als die des Vantage Point Tree. Zumindest bei der Betrachtung der ersten fünf Bilder sind einige ähnlich wie beim Goldstandard. Werden alle 20 gefundenen Objekte betrachtet, wird deutlich, dass der VP-Baum mehr gleiche Bilder gefunden hat. Wird diese Analyse auf mehrere Versuche mit unterschiedlichen Anfrageobjekten ausgedehnt, ist noch stärker erkennbar, dass die Bilder des VP-Baums zufriedenstellender sind.

Natürlich liegt es im Auge des Betrachters, was für ein gefundenes Objekt wichtig sein sollte: der Inhalt, die Farben oder die Formen. Rein subjektiv bewertet sind die Bilder natürlich nicht immer ähnlich im Hinblick auf den Inhalt des Bildes. Werden allerdings die Farbverteilungen und die Formen betrachtet, ist es verständlich, warum diese Bilder gefunden wurden. Aufgrund der Bildanalyse mit Hilfe des annularen Histogramms sind das genau die Aspekte, die bei dieser Untersuchung überprüft werden.

Letzendlich ist feststellbar, dass, wenn man sich nach dem Goldstandard richtet, der VP-Baum optisch bessere Ergebnisse liefert.



Abbildung 6.1: Anfragebild 1



Tabelle 6.1: Ergebnisse der Distanzmetrik mit sequentieller Suche

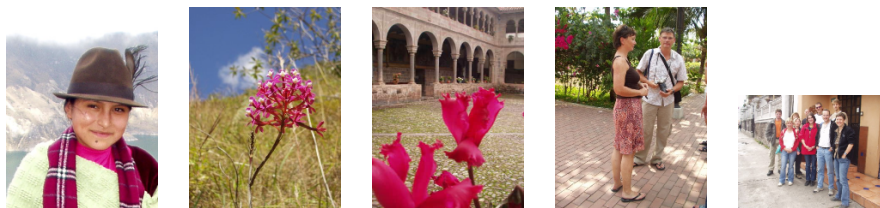


Tabelle 6.2: Ergebnisse des Evolving Tree

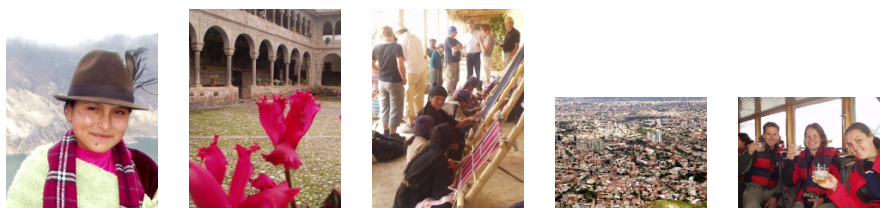


Tabelle 6.3: Ergebnisse des Vantage Point Tree



Abbildung 6.2: Anfragebild 2

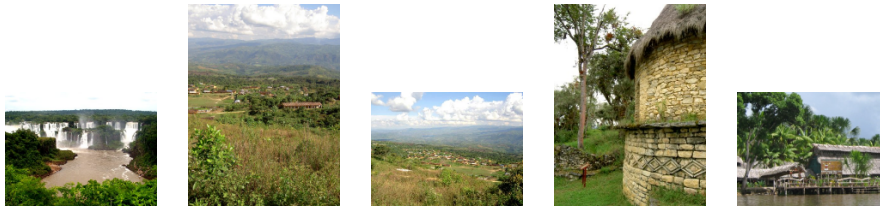


Tabelle 6.4: Ergebnisse der Distanzmetrik mit sequentieller Suche

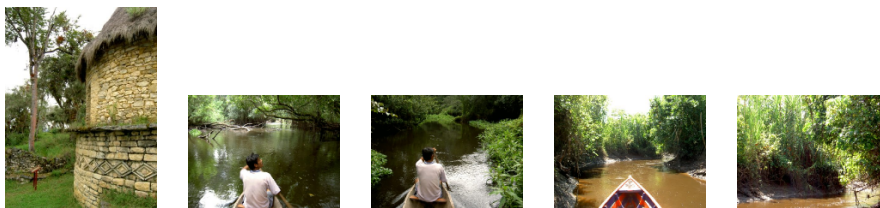


Tabelle 6.5: Ergebnisse des Evolving Tree

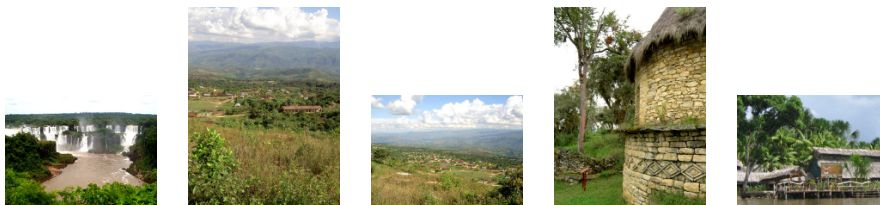


Tabelle 6.6: Ergebnisse des Vantage Point Tree



Abbildung 6.3: Anfragebild 3



Tabelle 6.7: Ergebnisse der Distanzmetrik mit sequentieller Suche



Tabelle 6.8: Ergebnisse des Evolving Tree



Tabelle 6.9: Ergebnisse des Vantage Point Tree

6.3 Maß der Precision

Unter Betrachtung der Voraussetzungen, dass beim Goldstandard die $Precision = 1$ ist und dass die Anzahl der betrachteten Objekte 20 ist, werden für die Indexstrukturen Evolving Tree und Vantage Point Tree die Genauigkeiten der gefundenen Bilder berechnet. Dazu wird die in Abschnitt 6.1 erläuterte Formel

$$Precision = \frac{|R \cap P|}{|P|}$$

genutzt.

Für die Suche aus dem Anfragebild 1 in Abbildung 6.1 erhält man als Ergebnis für den Evolving Tree $Precision_{ETree} = 0,6$ und für den Vantage Point Tree $Precision_{VPTree} = 0,7$.

Bei der Suche durch das Anfragebild 2 in Abbildung 6.2 ist das Ergebnis des Evolving Trees $Precision_{ETree} = 0,65$ und des VP-Baums $Precision_{VPTree} = 0,8$.

Bei der Berechnung der Precision für das Anfragebild 3 in Abbildung 6.3 lauten die Ergebnisse $Precision_{ETree} = 0,45$ und $Precision_{VPTree} = 0,8$.

Schon bei der Betrachtung dieser Ergebnisse wird deutlich, dass die Genauigkeit der Ergebnisse des Vantage Point Trees größer ist als die des Evolving Trees. Auch bei der Berechnung der Precision für viele verschiedene Anfragen wird dieser Trend bestätigt.

Bei einer Anzahl von 300 Beispielbildern und 100 Anfrageobjekten erhält man für den E-Baum für die Precision im Mittel 35% mit einer Standardabweichung von 14%. Der VP-Baum hat einen Precision-Mittelwert von 60% mit einer Standardabweichung von 12%. Erhöht man die Anzahl der Anfrageobjekte auf 200 bleiben diese Werte gleich.

Im Hinblick auf das Maß der Precision liegt der Vantage Point Tree eindeutig vor dem Evolving Tree. Die Ergebnisse des VP-Baums liegen in der Precision immer über 50%, wobei der E-Baum auch häufig Werte unter 50% aufweist.

7 Fazit

Um die Bildanalyse an sich aufzuwerten, könnten die einzelnen Ringe des gebildeten annularen Histogramms noch gewichtet werden. Die, vom Flächeninhalt aus betrachtet, kleineren, inneren Ringe haben die gleiche Bewertungsrelevanz wie die größeren Ringe. Dabei ist in den äußeren Ringen viel mehr Information enthalten, als in den inneren. Der Wichtungsfaktor müsste von innen nach außen größer werden, um die Analyse fairer zu gestalten.

Die unterschiedlichen Indexierungsverfahren müssten auf ihr Laufzeitverhalten getestet werden. Dabei sollten faire Grundbedingungen gelten. Das würde bedeuten, dass der Evolving Tree an die Bedingungen eines VP-Baums angepasst werden müsste. Demnach könnte das Verhalten der Indexe überprüft werden, in dem unterschiedliche Daten in diesen gespeichert werden.

Der Evolving Tree selbst müsste auf eine persistente Datenstruktur abgebildet werden, damit er eine bessere Kompression der Daten bereit stellt und eine höhere Laufzeitfähigkeit erhält. Dies könnte ein B-Baum sein. Durch eine Feineinstellung der gegebenen Parameter eines Evolving Trees kann das Endergebnis auch verbessert werden. Dazu muss genau analysiert werden, inwiefern die Parameter angepasst werden sollten.

Der Vantage Point Tree ist eine gute Indexstruktur. Aber auch dieses Verfahren kann noch deutlich verbessert werden. Es müsste untersucht werden, was die beste Methode wäre, um die Vantage Points zu ermitteln. Je genauer dieser Punkt definiert wird, umso besser werden die Mengen unterteilt und somit eine höhere Trefferquote ähnlicher Objekte gefördert.

Trotzdem bieten beide evaluierte Verfahren eine gute Grundlage zur Indexierung von komplexen Daten. Wenn die Verfahren noch optimiert werden, sind sie im Image Retrieval zur besseren Strukturierung und Laufzeitverbesserung einsetzbar.

Literaturverzeichnis

- [AR99] AIBING RAO, ROHINI K. SRIHARI, ZHONGFEI ZHANG: *Spatial Color Histograms for Content-Based Image Retrieval*, 1999.
- [AWcF00] FU, POLLY MEI-SHUEEN CHAN, YIN-LING CHEUNG UND YIU SANG MOON ADA WAI-CHEE: *Dynamic vp-tree indexing for n-nearest neighbor search given pair-wise distances*. The VLDB Journal, 2000.
- [Eib05] EIBL, PROFESSOR DR. MAXIMILIAN: *Vorlesung Medienergonomie*, 2005.
- [Eib06] EIBL, PROFESSOR DR. MAXIMILIAN: *Vorlesung Information Retrieval II: Multimedia Retrieval*, 2005/2006.
- [JP04] JUSSI PAKKANEN, JUKKA IIVARINEN UND ERKKI OJA: *The Evolving Tree - A Novel Self-Organizing Network for Data Analysis*. In: *Proceedings of IJCNN 2004*, Budapest, Hungary, 2004.
- [Mar07] MARKOV, IL'YA: *VP-tree: Content-Based Image Indexing*. In: *Proceedings of IJCNN 2004*, SYRCoDIS 2007, 2007.
- [Pak03] PAKKANEN, JUSSI: *The Evolving Tree, a new kind of self-organizing neural network*. In: *Proceedings of the Workshop on Self-Organizing Maps '03*, Kitakyushu, Japan, 2003.
- [SB98] SIGGELKOW, SVEN und HANS BURKHARDT: *Image retrieval based on colour and nonlinear texture invariants*, 1998.
- [uJI04] JUKKA IIVARINEN, JUSSI PAKKANEN UND: *A Novel Self-Organizing Neural Network for Defect Image Classification*. In: *Proceedings of IJCNN 2004*, Budapest, Hungary, 2004.
- [Yia93] YIANILOS, PETER N.: *Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces*. In: *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1993.

